Biodiversity Data Journal 13: e138257
doi: 10.3897/BDJ.13.e138257

Software Description

Pytaxon: A Python software for resolving and correcting taxonomic names in biodiversity data

Marco A. Proença Neto[‡], Marcos P. A. De Sousa[§,‡]

[‡] Centro Universitário do Estado do Pará, Belém, Brazil
[§] Laboratório de Computação Avançada para Biodiversidade (COMBIO). Museu Paraense Emílio Goeldi, Belém, Brazil

Corresponding author: Marcos P. A. De Sousa (msousa@museu-goeldi.br)

Academic editor: Zachary Foster

Received: 30 Sep 2024 | Accepted: 02 Jan 2025 | Published: 08 Jan 2025

Citation: Proença Neto MA, De Sousa MPA (2025) Pytaxon: A Python software for resolving and correcting taxonomic names in biodiversity data. Biodiversity Data Journal 13: e138257. https://doi.org/10.3897/BDJ.13.e138257

Abstract

Background

The standardisation and correction of taxonomic names in large biodiversity databases remain persistent challenges for researchers, as errors in species names can compromise ecological analyses, land-use planning and conservation efforts, particularly when inaccurate data are shared on global biodiversity portals.

New information

We present pytaxon, a Python software designed to resolve and correct taxonomic names in biodiversity data by leveraging the Global Names Verifier (GNV) API and employing fuzzy matching techniques to suggest corrections for discrepancies and nomenclatural inconsistencies. The pytaxon offers both a Command Line Interface (CLI) and a Graphical User Interface (GUI), ensuring accessibility to users with different levels of computing expertise. Tests on spreadsheets derived from datasets published in the Global Biodiversity Information Facility (GBIF) demonstrated its effectiveness in identifying and resolving taxonomic errors. By mitigating the propagation of inaccuracies from researchers' datasets to global biodiversity databases, pytaxon supports more reliable conservation decisions and robust scientific investigations. Its contributions enhance data integrity and promote informed biodiversity management in a rapidly evolving global environment.

Keywords

biodiversity informatics, taxonomy, scientific names, species name matching, data cleaning

Introduction

The scientific names of species and the taxonomic groups to which they belong are fundamental for the organisation and storage of the vast range of information on global biodiversity, functioning as crucial identifiers in this context (Grenié et al. 2022). Assigning a species identification to an organism is a fundamental activity in a wide range of disciplines, including ecology, land-use planning and conservation (Tyrrell 2019). Incorrect, ambiguous or synonymous taxon names present a fundamental problem for the study of comparative biology and directly affect the analysis of species occurrence data needed in ecological, biogeographical and fauna and flora conservation applications (Meyer et al. 2016, Thomson et al. 2021).

Traditionally, species data are often recorded in spreadsheets due to their ability to make the organisation and storage of information more flexible. These spreadsheets, frequently used by researchers, are a primary source for biodiversity databases (Güntsch et al. 2014 Bernardo et al. 2015, Silva et al. 2017, da Costa Prudente et al. 2019, da Silva et al. 2019). However, spreadsheets with large volumes of data are vulnerable to errors and difficult to correct (Broman and Woo 2018) and ensuring high quality species occurrence data in spreadsheets is a challenge (Foley 2010, Brown 2013, Magalhaes 2019). When incorrect data are included in a biological collections database, it can compromise the quality and reliability of scientific research (Moudrý and Devillers 2020). If such information is shared on international biodiversity information portals, the impact of erroneous data could expand globally (Saran et al. 2022).

Although renowned portals such as the Global Biodiversity Information Facility (GBIF; http://www.gbif.org) and the Atlas of Living Australia (ALA; https://www.ala.org.au) allow the import of spreadsheets with biodiversity data and have the ability to identify potentially incorrect or incomplete entries, corrections are not made automatically. It is the responsibility of the data provider to manually correct each identified error (Mesibov 2018). This correction process can become exhaustive when dealing with numerous spreadsheets, as each one needs to be individually reviewed.

Furthermore, there are online platforms that permit the search and resolution of taxonomic names in certain cases, such as the Global Names Verifier (GNV; https://

verifier.globalnames.org) and the Taxonomic Name Resolution Service (TNRS; https://tnrs.biendata.org). However, these platforms do not directly interface with spreadsheets and web-based queries are limited to a certain amount of scientific name entries. Ideally, researchers should have at their disposal a free tool that can interact with spreadsheets to validate taxonomic names in a simple and automated way, thereby facilitating the research process.

Currently, there is a variety of R packages available that assist in the process of standardising taxon names, including Taxadb (Norman et al. 2020), WorldFlora (Maitner et al. 2017, Kindt 2020) and U.Taxonstand (Zhang and Qian 2023). However, many researchers encounter difficulties in acquiring programming skills during their academic training (Emery et al. 2021), which can limit the adoption of these tools. This challenge underscores the need for user-friendly graphical interfaces and accessible methodologies to broaden the utilisation of such resources, as elaborated in the subsequent paragraph.

In many studies involving data analysis in biology, the prevalence of software operating with Command Line Interfaces (CLI) is evident, but this becomes a challenge in terms of accessibility, particularly for users with limited computing experience, as noted by Joppich and Zimmer (2019) and Chen et al. (2020). Graphical User Interfaces (GUIs) have emerged as a more convenient alternative, reducing errors in parameter settings and the overall usage burden. While a GUI does not inherently ensure a more user-friendly application, it does facilitate access and operation of the software, making it more accessible and less intimidating for users (Anslan et al. 2017).

To optimise and facilitate the review of taxonomic names in researchers' spreadsheets, we present pytaxon, a Python-based tool developed to identify and correct errors in species names within spreadsheets. The tool leverages the GNV API to compare user-provided data against trusted taxonomic name databases (https://verifier.globalnames.org/data_sources). Notably, pytaxon offers both a Command Line Interface (CLI) for automation and a Graphical User Interface (GUI) designed for users unfamiliar with programming, ensuring accessibility and usability for a broad audience.

Project description

Title: pytaxon

Design description: Pytaxon is developed in Python and designed to facilitate the validation and correction of taxonomic data in biodiversity spreadsheets. It utilises the Pandas library (Betancourt and Chen 2019) to manipulate and analyse CSV files and Excel spreadsheets, while integrating the GNV API through the requests library to access a comprehensive taxonomic database. The Fuzzy Matching functionality provided by the GNV API is leveraged by pytaxon to suggest corrections for naming discrepancies. Fuzzy Matching is a technique that identifies similarities between taxonomic names by calculating a similarity score to detect and correct typographical errors, improving taxonomic consistency and harmonisation in biodiversity datasets (Grenié et al. 2022). Additionally, the software uses the argparse library (https://pypi.org/project/argparse) for command-line operations to manage input parameters and arguments, making it a versatile tool for users familiar with Python pipelines.

The graphical user interface (GUI) is developed using Tkinter (https://docs.python.org/pt-br/3/library/tkinter.html), providing an interactive platform for users with minimal programming experience. It includes features for visualising data discrepancies and correcting taxonomic errors directly within the interface. Matplotlib is integrated for detailed visual data representation, enhancing user interaction. Pytaxon supports two ways of working: a command-line interface (CLI) for automation and a GUI for ease of use, ensuring accessibility to a broader range of users. The software is open-source and available on GitHub (https://github.com/pytaxon/pytaxon-cli) and PyPI (https://pypi.org/project/pytaxon), where the installation instructions and usage details can be found. This accessibility makes it easy to integrate, modify or use pytaxon for managing and correcting taxonomic data in large biodiversity datasets.

Web location (URIs)

Homepage: https://doi.org/10.5281/zenodo.14457929

Technical specification

Programming language: Python 3.10

Operational system: Windows, Linux

Usage licence

Usage licence: Other

IP rights notes: MIT Licence

Implementation

Implements specification

Below are the steps of pytaxon's workflow explained in detail; its visual workflow may be viewed at Fig. 1.

Loading the Original Spreadsheet

Loading the original spreadsheet is the first step in pytaxon's workflow, whether using the GUI or CLI. In the CLI, the command *-os* or *--original_spreadsheet* is used to provide the spreadsheet to be analysed, which must be in CSV or Excel format for compatibility. The command *-r* or *--columns* specifies the columns to be read, typically including "Kingdom", "Phylum", "Class", "Order", "Family", "Genus", "Species" and "ScientificName". Users can skip unnecessary columns by using the placeholder x (e.g. -r kingdom, phylum, class, order, family, x, x, scientificName), allowing flexibility in selecting relevant fields.

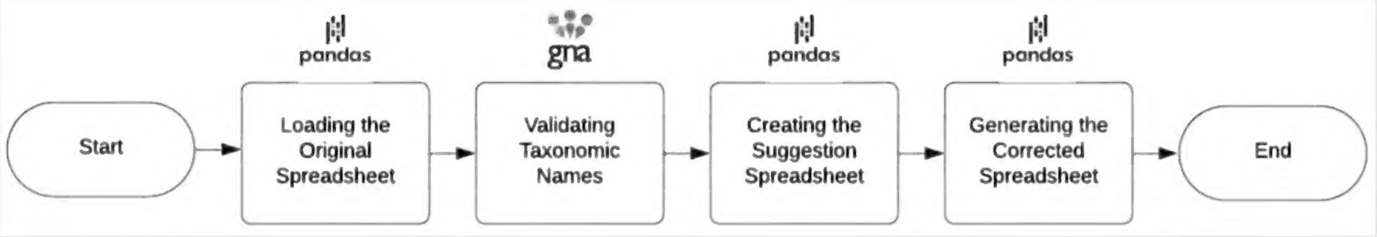

pandas
gna
pandas
pandas
Start
Loading the Original Spreadsheet
Validating Taxonomic Names
Creating the Suggestion Spreadsheet
Generating the Corrected Spreadsheet
End

Figure 1. doi

Pytaxon's workflow.

The *-si* or *--source_id* parameter allows the user to select the database which the programme will use to compare taxonomic names. The IDs for the supported data sources are: 1 for Catalog of Life (COL), 4 for National Center of Biotechnology Information (NCBI), 11 for Global Biodiversity Information Facility (GBIF) and 180 for iNaturalist. Additionally, the *-ss* or *--suggested_spreadsheet* command generates a spreadsheet containing suggested corrections for taxonomic name errors, helping users review and validate discrepancies efficiently.

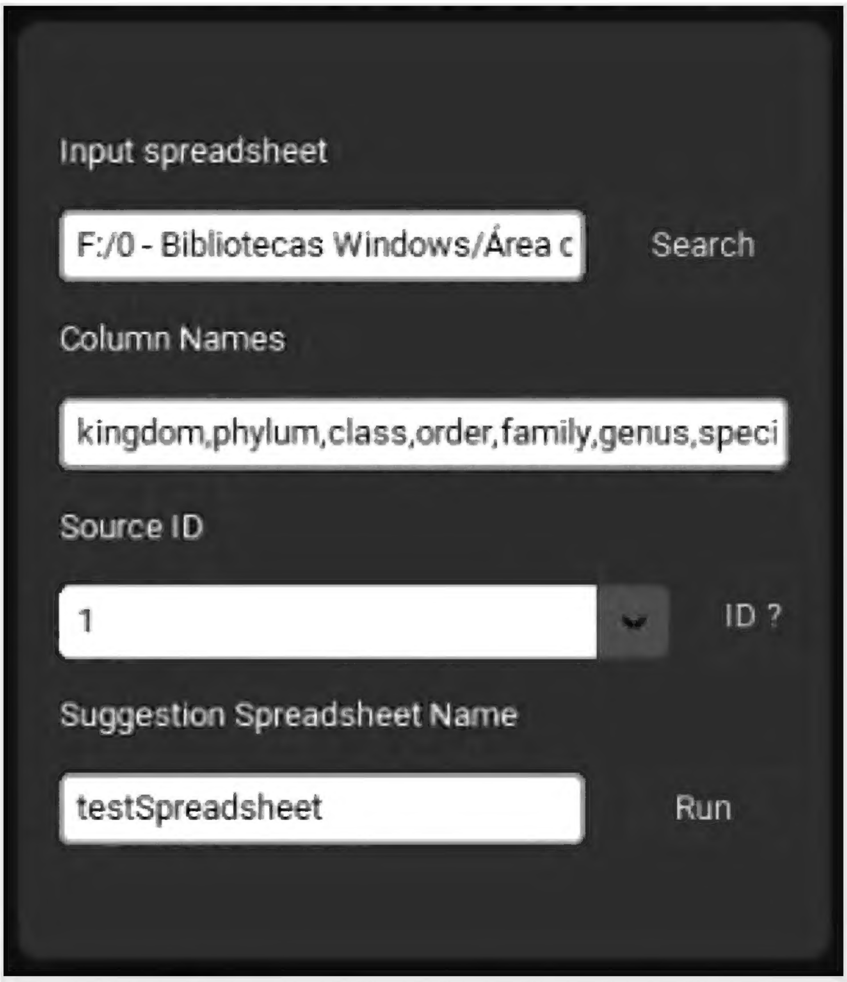
It is important to clarify that pytaxon focuses on validating and correcting taxonomic names within the dataset, rather than analysing species occurrence data. The frequency of taxonomic names in the dataset is used to identify discrepancies and suggest corrections.

For users who prefer a graphical interface, pytaxon also provides a GUI option for data entry, as shown in the attached Fig. 2. In the GUI, users can easily upload their input spreadsheet by selecting the file through a browse button. They can define the relevant column names directly in the interface, specify the source database ID from a dropdown menu and input a name for the suggestion spreadsheet. The GUI simplifies the process by providing an intuitive interface for data entry and validation.

Validating Taxonomic Names

Taxonomic name validation in pytaxon is achieved through integration with the Global Names Verifier (GNV) API (https://verifier.globalnames.org/api). This process begins with Exact Matching, where names are parsed to exclude authorship details and directly compared to entries in the GNV database. If no exact match is found, the API applies Fuzzy Matching, which corrects minor errors such as misspellings or formatting inconsistencies. These features ensure accurate name validation, even in datasets with imperfections, improving the reliability of biodiversity data. Fig. 3 shows the configuration file (config.json) that defines key parameters for the validation process.

The verify_taxon function, shown in Fig. 4, dynamically loads the config.json file to retrieve and apply the specified parameters. These include withAllMatches, which limits results to the best match to avoid ambiguity and withCapitalization, ensuring case-insensitive comparisons. The parameter withSpeciesGroup focuses the validation on species-level matches, while withUninomialFuzzyMatch activates fuzzy matching for genus-level names, enabling detection and correction of minor errors. The threshold for match acceptance is controlled by mainTaxonThreshold, balancing accuracy and tolerance for variations. Confidence metrics are included in the API response when withStats is enabled, helping to prioritise suggested corrections.



Input spreadsheet

F:/0 - Bibliotecas Windows/Área c

Column Names

kingdom,phylum,class,order,family,genus,speci

Source ID

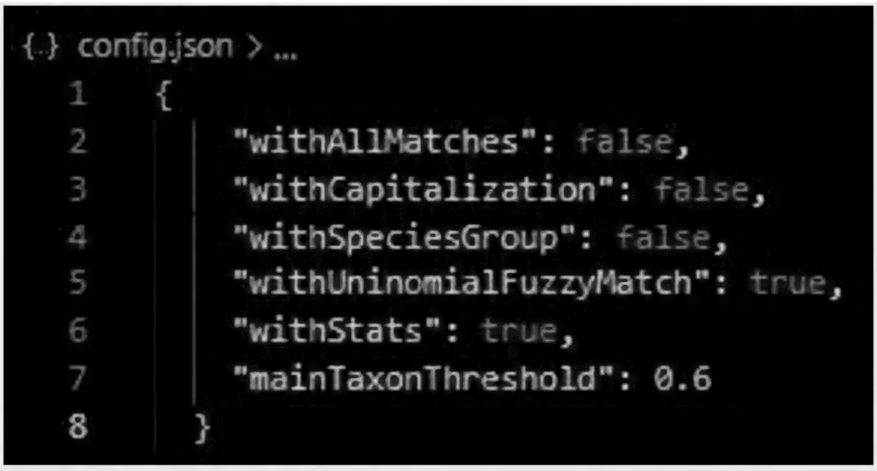
1

Suggestion Spreadsheet Name

testSpreadsheet

Figure 2. doi

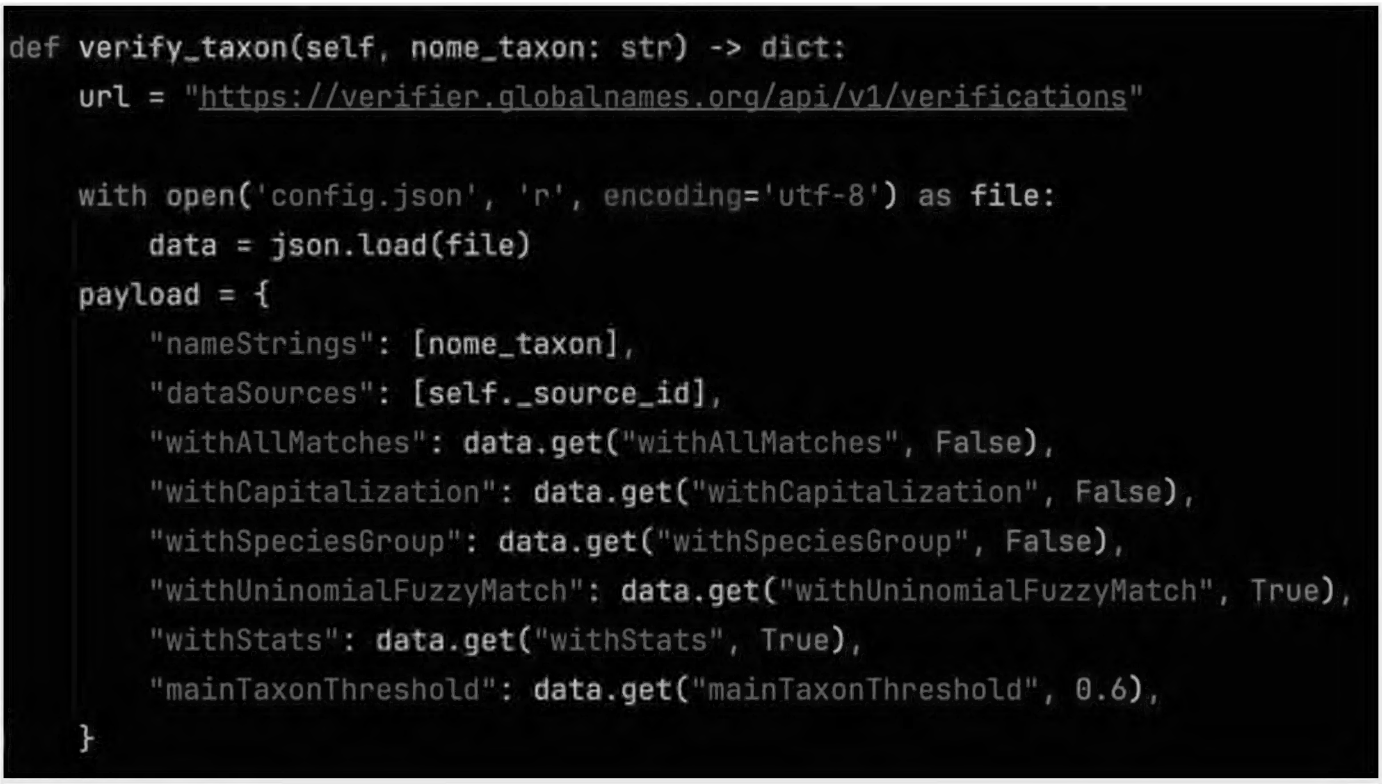
The pytaxon GUI displaying spreadsheet input fields and parameter selection options.



```
{
    "withAllMatches": false,
    "withCapitalization": false,
    "withSpeciesGroup": false,
    "withUninomialFuzzyMatch": true,
    "withStats": true,
    "mainTaxonThreshold": 0.6
}
```

Figure 3. doi

Configuration of the config.json file used in pytaxon.



```
def verify_taxon(self, nome_taxon: str) -> dict:
    url = "https://verifier.globalnames.org/api/v1/verifications"

    with open('config.json', 'r', encoding='utf-8') as file:
        data = json.load(file)
    payload = {
        "nameStrings": [nome_taxon],
        "dataSources": [self._source_id],
        "withAllMatches": data.get("withAllMatches", False),
        "withCapitalization": data.get("withCapitalization", False),
        "withSpeciesGroup": data.get("withSpeciesGroup", False),
        "withUninomialFuzzyMatch": data.get("withUninomialFuzzyMatch", True),
        "withStats": data.get("withStats", True),
        "mainTaxonThreshold": data.get("mainTaxonThreshold", 0.6),
    }
```

Figure 4. doi

The verify_taxon function that loads the config.json file in pytaxon.

Adjusting the values in config.json allows users to fine-tune the validation process according to their specific needs. For example, increasing mainTaxonThreshold improves precision, while enabling withAllMatches provides broader results. This flexibility, as illustrated in the attached figures, ensures that pytaxon is a customisable tool for standardising and validating taxonomic data in biodiversity datasets.

To address the challenges posed by the lack of standardised outputs from biodiversity database APIs, pytaxon processes and harmonises results from the four databases it currently supports: Catalog of Life, GBIF, NCBI and iNaturalist. This standardisation ensures consistency across outputs, enabling seamless validation and correction of taxonomic names and reducing potential errors arising from incompatible formats. By unifying these diverse outputs into a coherent format, pytaxon simplifies downstream analyses and enhances data reliability. In the future, we plan to extend support to additional databases, further increasing the software's versatility and applicability in biodiversity informatics.

Creating the Suggestion Spreadsheet

Based on the results of the Fuzzy Matching process provided by the GNV, pytaxon creates a suggestion spreadsheet with a name specified using the *-ss* or *--suggested_spreadsheet* command. This spreadsheet contains the rows with detected errors, the taxonomic rank (type), the presumably incorrect taxonomic names, the suggested corrections, the taxon source from the selected database and a "Change" column with options "*yes*" or "*no*" for the researcher to validate the corrections. This spreadsheet acts as a detailed report, highlighting areas that need attention and offering a clear and organised approach for rectifying inaccuracies in the dataset.

Generating the Corrected Spreadsheet

The final step in pytaxon's workflow is generating a corrected spreadsheet by applying the validated corrections from the suggested spreadsheet. This process updates the original dataset by replacing erroneous taxonomic names with accurate, user-validated ones. The corrected spreadsheet ensures data accuracy and standardisation, completing the workflow with a dataset of greater reliability.

Additional information

Results and conclusion

To demonstrate the advantages and flexibility of the pytaxon software package, we executed both CLI and GUI examples, highlighting its typical applications and core features through queries to analyse a given dataset and retrieve probable corrections. For these simulations, we inserted some errors into Uropygi dataset, which is openly available on the Zenodo platform (Sousa and Proença 2024), changing the Genus type "Thelyphonelus" to "Thelyphoneluss", the Order type "Uropygi" to "Urophygi" and the Kingdom type "Animalia" to "Aanimalia", then pytaxon was installed as a package from PyPI and its CLI commands were executed directly from the terminal with relevant queries as arguments, while the GUI version allowed for interactive engagement through its buttons and text blocks. This dual approach showcased the ease of use and comprehensive functionality of pytaxon in managing and correcting taxonomic data.

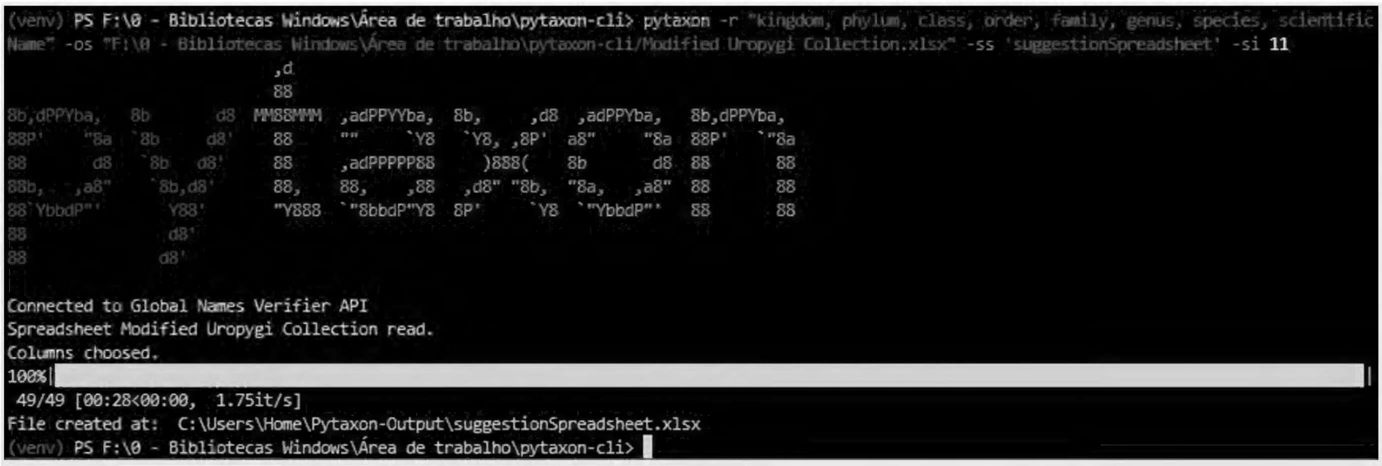
Command Line Interface Example

The command "*pytaxon -r 'kingdom, phylum, class, order, family, genus, species, scientificName' -os '<original spreadsheet path>/Modified Uropygi Collection.xlsx' -ss 'suggestionSpreadsheet' -si 11*" was used to begin the pytaxon analysing process, as shown in Fig. 5:

To initiate the pytaxon analysis, a dataset containing taxonomic information must be loaded using the appropriate CLI command. For example, the following command specifies the input spreadsheet, the columns to be analysed, the name of the suggestion spreadsheet and the source id:

pytaxon -os '<original spreadsheet path>/Modified Uropygi Collection.xlsx' -r 'kingdom, phylum, class, order, family, genus, species, scientificName' -ss 'suggestionSpreadsheet' -si 11

This command specifies the GBIF database for validation (-si 11), though users can select from other supported options such as the Catalog of Life, NCBI or iNaturalist. Once connected to the Global Names Verifier (GNV) API, the programme validates taxonomic names using exact matching to identify precise matches and Fuzzy Matching to address minor errors like misspellings. The generated suggestion spreadsheet, as illustrated in Fig. 6, highlights problematic entries and provides recommended corrections. A key feature of this spreadsheet is the "ID Source" column, which links each taxonomic name to its unique identifier in the reference database, ensuring traceability and reliability of the corrections. Additionally, the spreadsheet includes a "Change" column, allowing researchers to interactively review and decide whether to accept or reject each suggested correction, ensuring flexibility and control over the validation process.

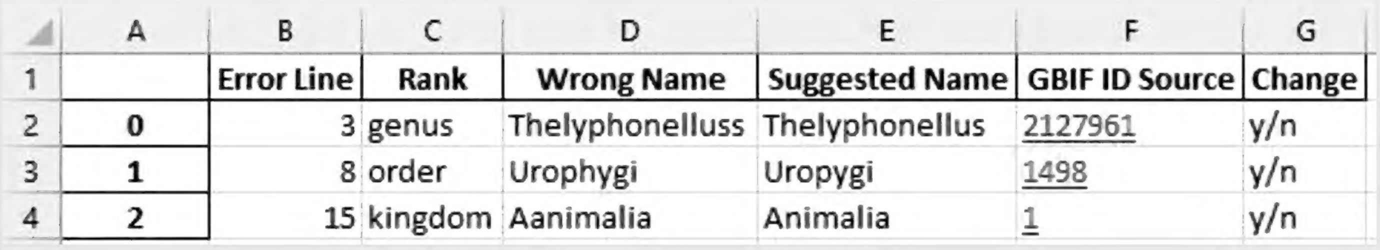


```
(venv) PS F:\0 - Bibliotecas Windows\Área de trabalho\pytaxon-cli> pytaxon -r "kingdom, phylum, class, order, family, genus, species, scientific
Name" -os "F:\0 - Bibliotecas Windows\Área de trabalho\pytaxon-cli/Modified Uropygi Collection.xlsx" -ss 'suggestionSpreadsheet' -si 11
                            ,d
                            88
8b,dPPYba,   8b       d8  MM88MMM  ,adPPYYba,  8b,      ,d8  ,adPPYba,   8b,dPPYba,
88P'    "8a  `8b     d8'    88     ""      `Y8  `Y8, ,8P'  a8"     "8a  88P'   `"8a
88       d8   `8b   d8'     88     ,adPPPPP88      )888(    8b       d8  88       88
88b,   ,a8"    `8b,d8'      88,    88,    ,88   ,d8" "8b,  "8a,   ,a8"  88       88
88`YbbdP"'       Y88'       "Y888  `"8bbdP"Y8  8P'      `Y8  `"YbbdP"'   88       88
88               d8'
88              d8'

Connected to Global Names Verifier API
Spreadsheet Modified Uropygi Collection read.
Columns choosed.
100%|████████████████████████████████████████████████████████████████████████████████|
 49/49 [00:28<00:00,  1.75it/s]
File created at:  C:\Users\Home\Pytaxon-Output\suggestionSpreadsheet.xlsx
(venv) PS F:\0 - Bibliotecas Windows\Área de trabalho\pytaxon-cli>
```

Figure 5. doi

The pytaxon CLI running on the Visual Studio Code terminal (Powershell) with the Uropygi dataset with inserted errors.



	A	B	C	D	E	F	G
1		Error Line	Rank	Wrong Name	Suggested Name	GBIF ID Source	Change
2	0	3	genus	Thelyphonelluss	Thelyphonellus	2127961	y/n
3	1	8	order	Urophygi	Uropygi	1498	y/n
4	2	15	kingdom	Aanimalia	Animalia	1	y/n

Figure 6. doi

The suggestion spreadsheet identifying errors and providing corrections.

After reviewing and validating the corrections in the suggestion spreadsheet, the user applies the changes to generate a cleaned dataset. This is achieved with the following command:

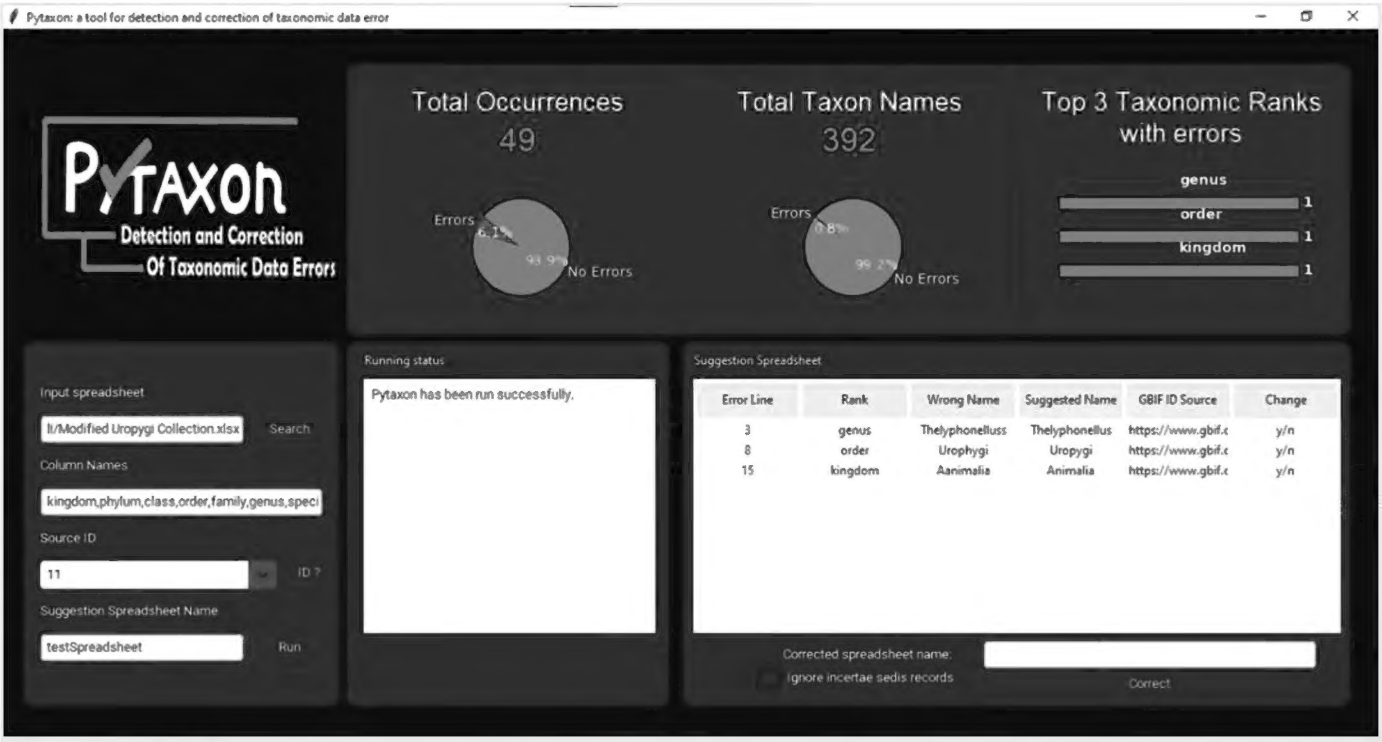
pytaxon -os '<original spreadsheet path>/Modified Uropygi Collection.xlsx' -ss '<suggestion spreadsheet path>/suggestionSpreadsheet.xlsx' -o 'correctedSpreadsheet'

This step creates a new corrected spreadsheet by replacing erroneous names with validated entries, ensuring a standardised and accurate dataset. By automating these processes, pytaxon streamlines the identification and correction of taxonomic errors, enhancing the quality and reliability of biodiversity data.

Graphical User Interface Example

The pytaxon graphical user interface (GUI), as shown in Fig. 7, provides a streamlined platform for validating and correcting taxonomic datasets. Its design is structured into functional components, each playing a vital role in the workflow. At the top of the interface, summary panels provide a comprehensive overview of the dataset. The "Total

Occurrences" panel displays the overall count of processed records, while the "Total Taxon Names" panel highlights the unique taxonomic names included in the dataset. The "Top 3 Taxonomic Ranks with Errors" panel focuses on the taxonomic levels most prone to discrepancies, helping users identify key areas for review.



Pytaxon: a tool for detection and correction of taxonomic data error

Total Occurrences
49

Total Taxon Names
392

Top 3 Taxonomic Ranks with errors

Input spreadsheet: lt/Modified Uropygi Collection.xlsx

Column Names: kingdom,phylum,class,order,family,genus,speci

Source ID: 11

Suggestion Spreadsheet Name: testSpreadsheet

Running status

Pytaxon has been run successfully.

Suggestion Spreadsheet

Error Line	Rank	Wrong Name	Suggested Name	GBIF ID Source	Change
3	genus	Thelyphonelluss	Thelyphonellus	https://www.gbif.c	y/n
8	order	Urophygi	Uropygi	https://www.gbif.c	y/n
15	kingdom	Aanimalia	Animalia	https://www.gbif.c	y/n

Corrected spreadsheet name:

Ignore incertae sedis records

Figure 7. doi

The pytaxon GUI application running with the Uropygi dataset with inserted errors.

The left-hand section of the interface is dedicated to data input and configuration. Here, users can upload their datasets via the "Input Spreadsheet" field, define the taxonomic columns to analyse, select the desired reference database (e.g. GBIF Backbone Taxonomy) and assign a name for the suggestion spreadsheet. Once these parameters are set, the "Run" button starts the analysis process, seamlessly transitioning to the execution phase.

As the analysis runs, the central "Running Status" panel dynamically updates to display progress and alert users to any issues encountered. This real-time feedback ensures users can monitor the process efficiently. On the right side, the "Suggestion Spreadsheet" section lists the detected errors, along with suggested corrections, in an interactive format. Users can review the discrepancies and directly accept or reject each correction within the interface, maintaining control over the final output.

With its integration of configurable input options, interactive error correction tools and live progress monitoring, the pytaxon GUI offers a robust and user-friendly environment for improving the quality of taxonomic datasets. This intuitive design ensures accessibility for users with varying levels of technical expertise, while maximising efficiency in the validation process.

Unit Testing in pytaxon

Developing unit tests for the pytaxon software was a important step in ensuring its robustness and reliability. The test suite, named "Test pytaxon" (De Sousa and Proença 2024) was implemented using python's unittest framework (https://docs.python.org/3/library/unittest.html). These tests evaluate critical functionalities, including taxonomic name validation, input file handling and the creation of corrected spreadsheets. This approach ensures that pytaxon performs reliably in both standard and edge-case scenarios.

The tests included specific scenarios to assess compatibility with various input formats, such as CSV and Excel files. Additionally, edge cases, such as empty files, blank lines, and missing columns, were simulated to verify the software's ability to effectively manage incomplete or inconsistent inputs. Clear error messages and integrated solutions were added to enhance user experience and confidence.

Comparing Pytaxon to Other Tools

The pytaxon stands out from other widely used tools, such as the Taxonomic Name Resolution Service (TNRS), Taxadb, WorldFlora and U.Taxonstand, due to its accessibility, flexibility and comprehensiveness. The TNRS is designed exclusively for the standardiation of plant taxonomic names and relies on specific databases like World Flora Online (WFO) and the World Checklist of Vascular Plants (WCVP). In contrast, pytaxon adopts a broader approach, enabling the validation of taxonomic names across a wide range of biological groups. It utilises diverse taxonomic sources, including the Catalog of Life, GBIF Backbone Taxonomy, iNaturalist and the National Center for Biotechnology Information (NCBI).

Another key advantage of pytaxon is its user-friendly interface. While tools like Taxadb and U.Taxonstand require programming skills, pytaxon provides both a Command Line Interface (CLI) and a Graphical User Interface (GUI), making it accessible to researchers with different levels of familiarity in computing. This integration allows for the direct manipulation of spreadsheets in CSV and Excel formats, enhancing efficiency in the processing and validation of taxonomic data.

Conclusions

The pytaxon provides an innovative solution to the persistent challenge of standardising and correcting taxonomic names in biodiversity data. By seamlessly integrating a graphical user interface with command-line functionality, the software is accessible to users with diverse technical skills. Leveraging exact and Fuzzy Matching techniques in conjunction with the GNV API, it efficiently detects and resolves taxonomic discrepancies in large datasets. This capability ensures data accuracy and streamlines the validation process for spreadsheets commonly used in biodiversity research. By enhancing the quality of taxonomic data, pytaxon contributes to more reliable ecological analyses, informed conservation planning and robust scientific investigations.

Acknowledgements

We thank the following institutions, which contributed to ensuring the success of our work: Museu Paraense Emílio Goeldi (MPEG) for the datasets for test cases and the support by Centro Universitário do Estado do Pará (CESUPA) with the PIBICT scientific initiation scholarship project funding.

References

- Anslan S, Bahram M, Hiiesalu I, Tedersoo L (2017) PipeCraft: Flexible open-source toolkit for bioinformatics analysis of custom high-throughput amplicon sequencing data. Molecular Ecology Resources 17 (6). https://doi.org/10.1111/1755-0998.12692
- Bernardo IR, Borges M, Baranauskas MCC, Santanchè A (2015) Interpretation of Construction Patterns for Biodiversity Spreadsheets. Lecture Notes in Business Information Processing397-414. https://doi.org/10.1007/978-3-319-22348-3_22
- Betancourt R, Chen S (2019) pandas Library. Python for SAS Users65-109. https://doi.org/10.1007/978-1-4842-5001-3_3
- Broman K, Woo K (2018) Data Organization in Spreadsheets. The American Statistician 72 (1): 2-10. https://doi.org/10.1080/00031305.2017.1375989
- Brown BV (2013) Automating the "Material examined" section of taxonomic papers to speed up species descriptions. Zootaxa 3683 (3). https://doi.org/10.11646/zootaxa.3683.3.8
- Chen C, Chen H, Zhang Y, Thomas H, Frank M, He Y, Xia R (2020) TBtools: An integrative toolkit developed for interactive analyses of big biological data. Molecular Plant 13 (8): 1194-1202. https://doi.org/10.1016/j.molp.2020.06.009
- da Costa Prudente AL, Ramos L, Silva T, Sarmento J, Dourado A, Silva F, Almeida P, Santos C, Sousa M (2019) Dataset from the snakes (Serpentes, Reptiles) collection of the Museu Paraense Emílio Goeldi, Pará, Brazil. Biodiversity Data Journal 7 https://doi.org/10.3897/bdj.7.e34013
- da Silva V, Aguiar-Neto M, Teixeira D, Santos C, de Sousa M, da Silva T, Ramos L, Bragio Bonaldo A (2019) Harvestmen occurrence database (Arachnida, Opiliones) of the Museu Paraense Emílio Goeldi, Brazil. Biodiversity Data Journal 7 https://doi.org/10.3897/bdj.7.e47456
- De Sousa M, Proença M (2024) Test pytaxon: Unit testing for the pytaxon software. Zenodo https://doi.org/10.5281/zenodo.14538755
- Emery NC, Crispo E, Supp SR, Farrell KJ, Kerkhoff AJ, Bledsoe EK, O'Donnell KL, McCall AC, Aiello-Lammens ME (2021) Data science in undergraduate life science education: A need for instructor skills training. BioScience 71 (12): 1274-1287. https://doi.org/10.1093/biosci/biab107
- Foley D (2010) A spreadsheet mapping approach for error checking and sharing collection point data. Biodiversity Informatics 7 (3). https://doi.org/10.17161/bi.v7i3.3986
- Grenié M, Berti E, Carvajal-Quintero J, Dädlow GML, Sagouis A, Winter M (2022) Harmonizing taxon names in biodiversity data: A review of tools, databases and best practices. Methods in Ecology and Evolution 14 (1): 12-25. https://doi.org/10.1111/2041-210x.13802

- Güntsch A, Mathew C, Obst M, Vicario S, Haines R, Williams A, de Jong Y, Goble C (2014) A semi-automated workflow for biodiversity data retrieval, cleaning, and quality control. Biodiversity Data Journal 2 https://doi.org/10.3897/bdj.2.e4221
- Joppich M, Zimmer R (2019) From command-line bioinformatics to bioGUI. PeerJ 7 https://doi.org/10.7717/peerj.8111
- Kindt R (2020) WorldFlora: An R package for exact and fuzzy matching of plant names against the World Flora Online taxonomic backbone data. Applications in Plant Sciences 8 (9). https://doi.org/10.1002/aps3.11388
- Magalhaes ILF (2019) Spreadsheets to expedite taxonomic publications by automatic generation of morphological descriptions and specimen lists. Zootaxa 4624 (1). https://doi.org/10.11646/zootaxa.4624.1.12
- Maitner B, Boyle B, Casler N, Condit R, Donoghue J, Durán S, Guaderrama D, Hinchliff C, Jørgensen P, Kraft NB, McGill B, Merow C, Morueta-Holme N, Peet R, Sandel B, Schildhauer M, Smith S, Svenning J, Thiers B, Violle C, Wiser S, Enquist B (2017) The bien r package: A tool to access the Botanical Information and Ecology Network (BIEN) database. Methods in Ecology and Evolution 9 (2): 373-379. https://doi.org/10.1111/2041-210x.12861
- Mesibov R (2018) An audit of some processing effects in aggregated occurrence records. ZooKeys 751: 129-146. https://doi.org/10.3897/zookeys.751.24791
- Meyer C, Weigelt P, Kreft H (2016) Multidimensional biases, gaps and uncertainties in global plant occurrence information. Ecology Letters 19 (8): 992-1006. https://doi.org/10.1111/ele.12624
- Moudrý V, Devillers R (2020) Quality and usability challenges of global marine biodiversity databases: An example for marine mammal data. Ecological Informatics 56 https://doi.org/10.1016/j.ecoinf.2020.101051
- Norman KA, Chamberlain S, Boettiger C (2020) taxadb: A high-performance local taxonomic database interface. Methods in Ecology and Evolution 11 (9): 1153-1159. https://doi.org/10.1111/2041-210x.13440
- Saran S, Chaudhary SK, Singh P, Tiwari A, Kumar V (2022) A comprehensive review on biodiversity information portals. Biodiversity and Conservation 31: 1445-1468. https://doi.org/10.1007/s10531-022-02420-x
- Silva TMd, Santos JCd, Ferreira VAV, Cruz Ramos LAd, Wosiacki WB, Sousa MPAd (2017) Data from the ichthyological collection of the Museu Paraense Emílio Goeldi. ZooKeys 687: 89-99. https://doi.org/10.3897/zookeys.687.11233
- Sousa M, Proença M (2024) Modified Uropygi collection dataset for pytaxon simulation. https://doi.org/10.5281/zenodo.14472861
- Thomson S, Thiele K, Conix S, Christidis L, Costello MJ, Hobern D, Nikolaeva S, Pyle R, van Dijk PP, Weaver H, Zachos F, Zhang Z, Garnett S (2021) Towards a global list of accepted species II. Consequences of inadequate taxonomic list governance. Organisms Diversity & Evolution 21 (4): 623-630. https://doi.org/10.1007/s13127-021-00518-8
- Tyrrell C (2019) A method to implement continuous characters in digital identification keys that estimates the probability of an annotation. Applications in Plant Sciences 7 (5). https://doi.org/10.1002/aps3.1247
- Zhang J, Qian H (2023) U.Taxonstand: An R package for standardizing scientific names of plants and animals. Plant Diversity 45 (1): 1-5. https://doi.org/10.1016/j.pld.2022.09.001